

Requested Patent: JP2003150442A

Title: MEMORY DEVELOPMENT PROGRAM AND DATA PROCESSOR ;

Abstracted Patent: JP2003150442 ;

Publication Date: 2003-05-23 ;

Inventor(s): TANABE YUSUKE ;

Applicant(s): FUJITSU LTD ;

Application Number: JP20010352816 20011119 ;

Priority Number(s): ;

IPC Classification: G06F12/02; G06F12/00; G06F17/30 ;

Equivalents: ;

**ABSTRACT:**

**PROBLEM TO BE SOLVED:** To develop tree structure data having already defined structure such as XML data in a memory at a high speed.**SOLUTION:** A structured memory 5 relating memory regions 5a, 5b, 5c to a tree structure based on a document structure definition file 1a is ensured in advance when, for example, starting the whole system and installing a memory development program 3 (step S1). When receiving a development demand concerning the tree structure data 1 transmitted from an application program 2 (step S2), element data included in the read tree structure data 1 is sequentially stored in the corresponding memory regions 5a, 5b, 5c in the structured memory 5 (step S3).

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-150442

(P2003-150442A)

(43) 公開日 平成15年5月23日 (2003.5.23)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 0 6 F 12/02	5 1 0	G 0 6 F 12/02	5 1 0 A 5 B 0 6 0
12/00	5 4 6	12/00	5 4 6 Z 5 B 0 7 5
	5 4 7		5 4 7 Z 5 B 0 8 2
	5 9 0		5 9 0
17/30	1 1 0	17/30	1 1 0 F

審査請求 未請求 請求項の数 5 O L (全 14 頁) 最終頁に続く

(21) 出願番号 特願2001-352816(P2001-352816)

(22) 出願日 平成13年11月19日 (2001.11.19)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(72) 発明者 棚辺 勇輔

富山県婦負郡八尾町保内二丁目2番1 株式会社富山富士通内

(74) 代理人 100092152

弁理士 服部 毅巖

Fターム(参考) 5B06D AA04 AA12 AA14 AC11

5B075 KK02 ND36

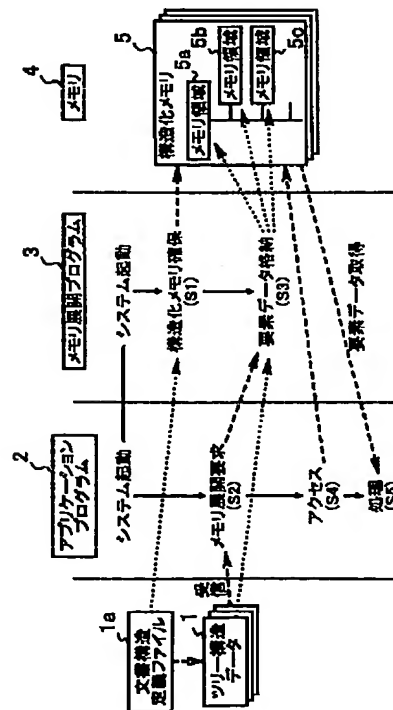
5B082 CA08

(54) 【発明の名称】 メモリ展開プログラムおよびデータ処理装置

(57) 【要約】

【課題】 XMLデータ等の構造定義済みのツリー構造データを高速にメモリ上に展開する。

【解決手段】 文書構造定義ファイル1aに基づいてメモリ領域5a、5bおよび5cをツリー構造に関連付けた構造化メモリ5を、例えばシステム全体の起動時やメモリ展開プログラム3のインストール時等にあらかじめ確保し(ステップS1)、アプリケーションプログラム2から送信されたツリー構造データ1についての展開要求を受信すると(ステップS2)、読み込んだツリー構造データ1に含まれる要素データを構造化メモリ5中の対応するメモリ領域5a、5bおよび5cに順次格納する(ステップS3)。



## 【特許請求の範囲】

【請求項1】 あらかじめ文書構造定義が行われたツリー構造データをメモリ上に展開する処理をコンピュータに実行させるメモリ展開プログラムにおいて、前記文書構造定義に基づいてメモリ領域をツリー構造に関連付けた構造化メモリをあらかじめ確保し、前記ツリー構造データについての展開要求を受信すると、読み込んだ前記ツリー構造データに含まれる要素データを前記構造化メモリ中の対応する前記メモリ領域に格納する、処理を前記コンピュータに実行させることを特徴とするメモリ展開プログラム。

【請求項2】 前記構造化メモリをあらかじめ複数確保しておき、前記展開要求を行うスレッドが複数存在する場合に、確保されたうちの前記要素データの格納が行われていないいずれかの前記構造化メモリを前記各スレッドが任意に使用可能とされることを特徴する請求項1記載のメモリ展開プログラム。

【請求項3】 前記構造化メモリに格納された前記要素データを保持しておく必要がなくなった場合は、前記展開要求に対する待機状態において、該当する前記構造化メモリを新たな前記ツリー構造データの格納処理が可能な空きメモリとして設定することを特徴とする請求項1記載のメモリ展開プログラム。

【請求項4】 前記文書構造定義において出現回数が不特定な要素が設定されている場合に、前記ツリー構造データの前記構造化メモリへの格納処理を行うたびに前記要素の実際の出現回数を記憶して順次蓄積し、蓄積した値に基づいて必要な前記メモリ領域の数を統計的に算出し、前記展開要求に対する待機状態において、確保している前記構造化メモリの構造を前記算出値に応じて変更することを特徴とする請求項1記載のメモリ展開プログラム。

【請求項5】 あらかじめ文書構造定義が行われたツリー構造データを使用して処理するデータ処理装置において、前記ツリー構造データが展開されるメモリと、前記文書構造定義に基づいてツリー構造に関連付けたメモリ領域を前記メモリ上にあらかじめ確保し、前記ツリー構造データについての展開要求を受信すると、読み込んだ前記ツリー構造データに含まれる要素データを対応する前記メモリ領域に格納するメモリ展開手段と、使用する前記ツリー構造データに対する前記展開要求を前記メモリ展開手段に送信し、前記要素データが格納された前記メモリ領域にアクセスすることにより前記ツリー構造データを読み取って所定の処理を行うツリー構造データ処理手段と、を有することを特徴とするデータ処理装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、あらかじめ文書構造定義が行われたXMLデータ等のツリー構造データをメモリ上に展開するためのメモリ展開プログラム、およびメモリ展開方法に関する。

## 【0002】

【従来の技術】近年、コンピュータシステムやコンピュータネットワーク間で使用される文書やデータに対する互換性に優れた構造化文書が注目を集めている。特に、SGML (Standard Generalized Markup Language) の文法定義を簡略化して処理パフォーマンスを向上し、ネットワーク適応性を向上したデータ記述言語であるXML (Extensible Markup Language) を利用した文書は、例えばB2B (Business to Business) といわれる企業間の電子取引の分野等で盛んに利用されている。このXMLでは、DTD (Document Type Definition: 文書型定義) あるいはXMLスキーマ等を用いて文書型を定義し、文書の構造化が図られる。

【0003】ところで、例えば企業間取引を運用するため等のアプリケーションプログラムがXML文書を取り扱うためには、XMLデータをメモリ上に展開して、XMLに従ってタグ付けされた文書をアプリケーションプログラムが解釈することを可能にするプログラムが必要となる。このようなプログラムは例えばDLL (Dynamic Link Library) ファイル等として提供され、アプリケーションプログラムからの要求に応じて、読み込んだXMLデータをツリー構造としてメモリ上に展開する。そして、アプリケーションプログラムは、このツリー構造にアクセスすることによりデータの利用が可能になる。

【0004】ここで、図11は、上記のようなメモリ展開プログラムによる従来のメモリ展開処理を示すフローチャートである。図11のフローチャートの開始時において、メモリ展開プログラムと、これを利用するアプリケーションプログラム等の利用側プログラムとが起動され、例えばネットワークや記録装置等からXMLデータがこのシステムに渡される。ステップS1101において、利用側プログラムよりXMLデータのメモリ展開要求が行われ、メモリ展開プログラムはこれを受信してメモリ展開処理の実行を開始する。

【0005】ステップS1102において、XMLデータを解析してルート要素を取り出す。ステップS1103において、取り出したルート要素を格納するためのメモリを確保する。ステップS1104において、確保したメモリにルート要素のデータを設定する。

【0006】ステップS1105において、XMLデータからルート要素以下の要素を順に1つ取り出す。ステップS1106において、取り出した要素のデータを格納するためのメモリを確保する。ステップS1107において、確保したメモリに要素のデータを設定する。ステップS1108において、この要素のツリーの親の要素との関連付けを行う。

【0007】ステップS1109において、読み込んだXMLデータ中のすべての要素のデータについてのメモリ展開が終了したか否かを判断し、終了した場合はステップS1110に進む。また、終了していない場合は、ステップS1105に戻り、再び要素を取り出してメモリに設定し、親との関連付けを行う処理を繰り返す。ステップS1110において、すべての要素のデータが展開され、展開済みのメモリを利用側プログラムに対して返却する。

【0008】次に、図12は、上記のメモリ展開処理により確保されるメモリの状態遷移を模式的に示す図である。図12(A)は、メモリ展開処理が開始されて、読み込まれたXMLデータからルート要素が取り出された状態を示しており、図11のステップS1102に対応する。この状態では、ルート要素に対するメモリ領域が確保されただけで、データは設定されていない。次に、(B)に示すように、確保されたメモリにルート要素のデータが設定される(ステップS1103に対応)。

【0009】その後、XMLデータより次の要素が1つ取り出され、(C)に示すように、取り出された要素に対するメモリ領域が確保される(ステップS1106に対応)。この状態では、新たに確保されたメモリは上記の(A)と同様に空の状態である。次に、(D)に示すように、確保されたメモリに要素のデータが設定される(ステップS1107に対応)。次に、(E)に示すように、データが設定されたメモリと、ルート要素のデータが格納されたメモリとの関連付けが行われ、各メモリ間のツリー構造における親子関係が定義される(ステップS1108に対応)。以降、XMLデータから要素が取り出されるたびに、メモリの確保とデータの設定、および関連付けが行われて、構造化されたデータが順次メモリに展開される。

#### 【0010】

【発明が解決しようとする課題】ところで、XMLのようなデータを構造化する言語は、タグを自由に設定できる等、ユーザにとって取り扱いやすいという特徴を有する反面、データのオーバーヘッドが大きいことから、アプリケーションプログラムによってXML文書を扱う際の処理負担が大きく、処理速度が遅くなることが従来から問題となっていた。

【0011】このうち、上述した従来のメモリ展開処理においては、XMLデータからツリー構造の要素を取り出すごとに、メモリ確保やデータ設定、および関連付けの各処理が行われるため、利用側プログラムからメモリ展開が要求された時点から結果返却までの処理時間が長くなってしまふ。この処理時間は、取り扱うデータ数が増加するほど長くなり、例えば、DTDにおける定義で出現回数があらかじめ特定されない要素では、データの出現に応じて逐次メモリの確保等が行われ、処理時間が大きく変動してしまう。

【0012】また、例えば、利用側プログラムがマルチスレッドで動作する場合では、各スレッドから要求に応じてそのたびにメモリ確保等の処理を行うため、一層処理効率が低い。

【0013】さらに、上記のメモリ展開プログラムでは、利用側プログラムからのメモリ展開要求に対する待機状態では動作していないため、CPU(Central Processing Unit)等のシステムリソースが有効に利用されているとは言えなかった。

【0014】本発明はこのような課題に鑑みてなされたものであり、XMLデータ等の構造化済みのツリー構造データを高速にメモリ上に展開することが可能なメモリ展開プログラムを提供することを目的とする。

【0015】また、本発明の他の目的は、XMLデータ等の構造化済みのツリー構造データを高速にメモリ上に展開することが可能なメモリ展開方法を提供することである。

#### 【0016】

【課題を解決するための手段】本発明では上記課題を解決するために、図1に示すような処理をコンピュータに実行させるためのメモリ展開プログラム3が提供される。このメモリ展開プログラム3は、あらかじめ文書構造化定義が行われたツリー構造データ1をメモリ4上に展開するためのプログラムであり、前記文書構造化定義に基づいてメモリ領域5a、5bおよび5cをツリー構造に関連付けた構造化メモリ5をあらかじめ確保し(ステップS1)、前記ツリー構造データ1についての展開要求を受信すると(ステップS2)、読み込んだ前記ツリー構造データ1に含まれる要素データを前記構造化メモリ5中の対応する前記メモリ領域5a、5bおよび5cに格納する(ステップS3)ことを特徴とする。

【0017】このような処理により、コンピュータはツリー構造データ1をメモリ5上に展開する装置として機能する。この装置では、ツリー構造データ1に対するメモリ5への展開要求に先立ち、ツリー構造データ1の要素データを格納するためのメモリ領域5a、5bおよび5cがあらかじめ確保される。このとき、例えば文書構造化定義ファイル1a等によりあらかじめ規定された文書構造化定義に基づいて、メモリ領域5a、5bおよび5cをツリー構造に関連付けた構造化メモリ5として確保される。この後、展開要求が受信されると、読み込まれた要素データが構造化メモリ5中の対応するメモリ領域5a、5bおよび5cに格納されることにより、ツリー構造データ1がメモリ5に展開される。従って、メモリ5に対する展開処理時には、メモリ領域5a、5bおよび5cを確保する処理やそれらをツリー構造に関連付ける処理が不要になり、処理負担が軽減される。

【0018】また、本発明では、あらかじめ文書構造化定義が行われたツリー構造データを使用して処理するデータ処理装置において、前記ツリー構造データが展開され

るメモリと、前記文書構造定義に基づいてツリー構造に関連付けたメモリ領域を前記メモリ上にあらかじめ確保し、前記ツリー構造データについての展開要求を受信すると、読み込んだ前記ツリー構造データに含まれる要素データを対応する前記メモリ領域に格納するメモリ展開手段と、使用する前記ツリー構造データに対する前記展開要求を前記メモリ展開手段に送信し、前記要素データが格納された前記メモリ領域にアクセスすることにより前記ツリー構造データを読み取って所定の処理を行うツリー構造データ処理手段と、を有することを特徴とするデータ処理装置が提供される。

【0019】このようなデータ処理装置では、ツリー構造データ処理手段から送信される、ツリー構造データに対するメモリへの展開要求に先立って、メモリ展開手段により、ツリー構造データの要素データを格納するためのメモリ領域があらかじめ確保される。このとき、あらかじめ規定された文書構造定義に基づいて、メモリ領域をツリー構造に関連付けた構造化メモリとして確保される。この後、展開要求が受信されると、読み込まれた要素データが構造化メモリ中の対応するメモリ領域に格納されることにより、ツリー構造データがメモリに展開される。従って、メモリに対する展開処理時には、メモリ領域を確保する処理やそれらをツリー構造に関連付ける処理が不要になり、処理負担が軽減される。

【0020】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して説明する。図1は、本発明の原理を説明するための原理図である。以下、図1を用いて本発明の概要を説明する。

【0021】図1に示すように、本発明は主に、ツリー構造データ1を利用して種々の処理を行うアプリケーションプログラム2と、このアプリケーションプログラム2が利用可能なようにツリー構造データ1をメモリ4上に展開するメモリ展開プログラム3とを実行可能なコンピュータ装置等の情報処理装置において実現される。

【0022】ツリー構造データ1は、例えばXML等に従って構造化されたデータであり、例えばDTD等の文書構造定義ファイル1aによりあらかじめ定義されたツリー構造を有している。文書構造定義ファイル1aは、アプリケーションプログラム2によって利用されるためのデータ型を定義している。このツリー構造データ1は、例えば図示しないネットワークを通じて受信したり、あるいは図示しないHDD（ハードディスクドライブ）等の記憶装置から読み出す、図示しない入力装置から入力される等により供給される。

【0023】アプリケーションプログラム2およびメモリ展開プログラム3は、通常は例えばHDD等の記憶装置に格納され、メモリ4等に読み出されて図示しないCPU（Central Processing Unit）等の制御装置によって実行される。

【0024】アプリケーションプログラム2は、文書構造定義ファイル1aに従ってフォーマットされたツリー構造データ1の供給を受けて、このツリー構造データ1の要素データを取り出して利用することで所定の処理を行う。例えば、ネットワークを通じて企業間の商取引を行う際に、ツリー構造データ1として受信した電子文書を認識して商品を受注を管理する等の処理を行う。このツリー構造データ1が供給されると、アプリケーションプログラム2は、メモリ展開プログラム3に対してツリー構造データ1のメモリ4への展開要求を行い、メモリ4上に展開されたデータにアクセスすることによって、ツリー構造データ1を認識してその要素データを利用することが可能となる。

【0025】メモリ展開プログラム3は、例えば、OS（Operating System）に組み込まれたDLLファイル等として提供され、アプリケーションプログラム2に利用される。このメモリ展開プログラム3は、アプリケーションプログラム2を含むシステム起動時、あるいはアプリケーションプログラム2の起動時等に起動されて、後述するように構造化メモリ5の確保等の初期処理を行った後、アプリケーションプログラム2からのメモリ4に対する展開要求に応じて、ツリー構造データの要素データをメモリ4上に展開し、展開済みメモリをアプリケーションプログラム2に返す処理を行う。

【0026】メモリ4は、例えば半導体によってなるRAM（Random Access Memory）であり、ツリー構造データ1が展開されて、アプリケーションプログラム2からのアクセスを可能にするとともに、例えば、アプリケーションプログラム2およびメモリ展開プログラム3の実行領域としても機能する。

【0027】次に、ツリー構造データ1のメモリ4への展開処理について説明する。上述したように、メモリ展開プログラム3は、システム起動時等の初期処理として、構造化メモリ5を確保する処理を行う（ステップS1）。この構造化メモリ5は、ツリー構造データに含まれる要素データを格納するメモリ領域5a、5bおよび5cを、ツリー構造をなすように関連付けて確保したものである。

【0028】ここで、供給されるツリー構造データ1のデータ構造は、文書構造定義ファイル1aによってあらかじめ定義されていることから、メモリ展開プログラム3は、この文書構造定義ファイル1aを参照して、確保するメモリ領域5a、5bおよび5cの有するツリー構造を設定する。これにより、供給されるツリー構造データ1のデータ構造にはほぼ一致した構造化メモリ5が確保される。なお、このとき各メモリ領域5a、5bおよび5cは単に確保されるだけで、データは空の状態である。

【0029】この後、アプリケーションプログラム2の動作上でツリー構造データ1を参照する段階になると、

アプリケーションプログラム2からこのツリー構造データ1のメモリ5上への展開要求が送信される(ステップS2)。メモリ展開プログラム3は、この展開要求を受信すると、ツリー構造データ1における要素データを、すでに確保した構造化メモリ5中の各メモリ領域5a、5bおよび5cに格納する処理を行う(ステップS3)。メモリ展開プログラム3は、供給されたツリー構造データ1のコードを順次解析し、このツリー構造に対応するように、確保されている構造化メモリ5中のメモリ領域5a、5bおよび5cに対して要素データを格納していく。1つの構造化メモリ5に対する要素データの格納が終了すると、展開したメモリ領域5a、5bおよび5cの情報をアプリケーションプログラム2に返却する。

【0030】その後、アプリケーションプログラム2は、要素データが格納された構造化メモリ5にアクセスして(ステップS4)、目的とする要素データを順次取得し、所定の処理を行う(ステップS5)。

【0031】以上のメモリ4への展開処理では、アプリケーションプログラム2からツリー構造データ1に対するメモリ5への展開要求が行われる前に、ツリー構造データ1の要素データを格納するためのメモリ領域5a、5bおよび5cがあらかじめ確保される。このとき、あらかじめ規定された文書構造定義に基づき、各要素データを格納するメモリ領域5a、5bおよび5cをツリー構造に関連付けた構造化メモリ5として確保される。従って、この後にアプリケーションプログラム2からの展開要求が受信されたときは、ツリー構造データ1から要素データが抽出され、構造化メモリ5中の対応するメモリ領域5a、5bおよび5cに格納される処理のみが行われる。従って、メモリ4への展開処理時における処理負担が軽減され、アプリケーションプログラム2からの展開要求に対して高速に処理することが可能となる。

【0032】なお、メモリ4への展開要求に先立って、構造化メモリ5を複数確保するようにしてもよい。例えば、アプリケーションプログラム2がマルチスレッドで動作するような場合には、確保された複数の構造化メモリ5のうち、要素データが展開されていないものを、各スレッドが任意に使用可能なようにしておく。これにより、あらかじめ確保しておいた構造化メモリ5を各スレッドが使い回すことができ、各スレッドの実行ごとに構造化メモリ5を確保する必要がなくなるので、展開処理を高速で行うことができる。また、確保される複数の構造化メモリ5は、異なる構造を定義した文書構造定義ファイル1aに応じて生成されてもよい。

【0033】また、メモリ4への展開処理が終了した後の、アプリケーションプログラム2からの展開要求に対する待機状態において、メモリ展開プログラム3は、要素データが格納された構造化メモリ5のうち、アプリケーションプログラム2からの使用が終了して、格納され

た要素データを保持しておく必要がなくなった構造化メモリについて、空きメモリとして設定するようにしてもよい。

【0034】この場合、メモリ展開プログラム3は、構造化メモリ5中に格納された要素データについて、アプリケーションプログラム2による使用が終了したか否かについて常に監視し、使用が終了した構造化メモリ5については、展開要求の待機状態において、各メモリ領域5a、5bおよび5cに格納された要素データをリフレッシュする。または、リフレッシュを行わずに、該当する構造化メモリが新たなツリー構造データ1の格納が可能となったことを認識しておくようにしてもよい。このような処理により、確保しておいた構造化メモリ5を効率よく使用できるとともに、展開処理を行っていない期間を利用して処理を行うので、システムリソースを有効に利用することが可能となる。

【0035】さらに、確保した構造化メモリ5にすべて要素データが格納され、空きがなくなった状態となった場合には、メモリ展開プログラム3はこのことを検知し、新たな構造化メモリ5を確保する処理を、展開要求に対する待機状態において行うようにしてもよい。これにより、次にアプリケーションプログラム2からの展開要求を受信した際に、あらかじめ構造化メモリ5の確保を行わずに済み、高速に処理することが可能となる。

【0036】また、文書構造定義ファイル1aでは、ツリー構造上のある要素に対してその子要素の出現回数を特定しないことができる。このように設定されていた場合、メモリ展開プログラム3は、文書構造定義ファイル1aを参照しても、確保する構造化メモリ5のツリー構造が、実際に供給されるツリー構造データ1の構造と一致するとは限らず、メモリ領域が足りない場合は、要素データの格納時にあらかじめメモリ領域5a、5bおよび5cを追加確保し、親要素との関連付けを行う必要が生じる。

【0037】そこで、このような場合は、例えば、ツリー構造データ1の構造化メモリ5への格納処理を行うたびに、該当する要素の実際の出現回数を記憶して順次蓄積しておく。そして、蓄積した値に基づいて統計値を算出し、この要素の出現回数を予想して必要なメモリ領域5a、5bおよび5cの数を決定し、確保している構造化メモリ5におけるツリー構造を順次変更可能なようにしておく。また、ツリー構造の変更は、アプリケーションプログラム2からの展開要求に対する待機状態において行う。これにより、出現回数が不特定な要素に対して、必要なメモリ領域5a、5bおよび5cの数を予想してあらかじめ確保するので、展開要求後にあらかじめメモリ領域5a、5bおよび5cを追加して確保する確率が低減され、全体としてメモリ4へ展開する処理時間が短縮される。

【0038】次に、本発明の具体的な適用例について説

明する。なお、以下の実施の形態例では、上記のツリー構造データ1の例としてXMLデータを使用し、また、文書構造定義ファイル1aの例としてDTDを使用することにする。

【0039】図2は、本発明を適用可能な商品販売システムのシステム構成例を示す図である。図2に示す商品販売システム100は、一般のユーザに対してネットワーク200を通じて商品を販売するためのシステムであり、例えば商品の販売元が商品を受注するための商品受注サーバ10と、一般のユーザに対して商品を販売し、販売された商品を販売元に発注する商品販売サーバ21および22と、一般のユーザによって商品の購入が行われる端末装置31、32および33によって構成される。これらの商品受注サーバ10、商品販売サーバ21および22、端末装置31、32および33は、ネットワーク200によって接続されており、このネットワーク200は、例えばインターネット等のような広域ネットワークである。

【0040】商品受注サーバ10は、例えばメーカ等の商品の販売元によって運用される情報処理装置であり、商品販売サーバ21および22において販売された商品について、その商品を出荷するための注文を受ける。受注のために商品販売サーバ21および22から受信するデータは、所定のDTDファイルによって定義されて統一された文書型を有するXMLデータとされる。商品受注サーバ10では、受信したXMLデータを基に商品の受注を管理するための受注管理プログラムと、XMLデータをメモリ上で展開して受注プログラムからのアクセスを可能にするメモリ展開プログラムとが実行される。

【0041】商品販売サーバ21および22は、端末装置31、32および33のユーザに対して商品を販売するための情報処理装置であり、例えば商品販売を行うWebサイトを提供する。また、これとともに、購入された商品を商品受注サーバ10へ発注するクライアントとしても機能する。商品販売サーバ21および22では、例えば商品販売用のWebサイトのファイルを端末装置31、32および33に送信して、返信されたデータを認識して電子決済等の処理を行うとともに、返信されたデータを基にして、DTDファイルにより定義された文書型に商品名や個数、購入者情報等の商品発注のためのデータを格納したXMLデータを生成し、商品受注サーバ10に対して送信する処理を行うプログラムが実行される。

【0042】このような商品販売サーバ21および22が提供する商品販売用のWebサイトは、例えば様々な商品を一括して取り扱うポータルサイトとされてもよい。また、図2では商品販売サーバ21および22の2つのみ示されているが、これに限らず複数設置されてもよい。さらに、例えば商品販売サーバ21および22において取り扱われる商品の販売元ごとに、商品受注サーバ

10が複数設定されてもよい。

【0043】端末装置31、32および33は、商品販売サーバ21および22を通じて商品の購入を申し込むための情報処理装置である。例えば、商品販売サーバ21および22によって商品販売用のWebサイトが提供されている場合、端末装置31、32および33ではこのWebサイトを閲覧するためのブラウザプログラム等が実行される。端末装置31、32および33のユーザは、商品販売用のWebサイトにアクセスし、商品を選択して購入を申し込むことができ、また電子決済等を行ってもよい。

【0044】この端末装置31、32および33は、例えば、ユーザの家庭等に設置されるPC（パーソナルコンピュータ）や、あるいは店頭や街頭に設置される情報端末として実現される。なお、図2では端末装置31、32および33の3つのみ示したが、実際には多数存在してもよい。

【0045】このような商品販売システム100では、端末装置31、32および33のユーザが、商品受注サーバ10側で受注される商品を購入する際に、商品販売サーバ21または22を通じて注文を行う。この際にネットワーク200を通じて送受信されるデータとしては、商品販売サーバ21および22から商品受注サーバ10に対してのみ、DTDファイルで文書構造が定義されたXMLデータという統一されたフォーマットが用いられればよい。

【0046】次に、図3は、本発明の実施の形態に用いる商品受注サーバ10の内部構成例を示すブロック図である。図3に示すように、商品受注サーバ10は、CPU11、RAM12、HDD13、グラフィック処理部14、入力I/F（インタフェース）15および通信I/F16によって構成され、これらはバス17を介して相互に接続されている。

【0047】CPU11は、商品受注サーバ10の全体に対する制御をつかさどる。RAM12はCPU11に実行させるプログラムの少なくとも一部や、このプログラムによる処理に必要な各種データを一時的に記憶する。本発明では、例えば、ネットワーク200を通じて受信されたXMLデータ13cを記憶して展開するためにツリー構造化された構造化メモリの領域が、RAM12上に設けられる。

【0048】HDD13には、OSやアプリケーションプログラム、各種データが格納される。本発明では主に、商品を受注してその受注状況を管理する受注管理プログラム13aや、OSに組み込まれたDLLファイルとして存在するメモリ展開プログラム13b、ネットワーク200を通じて受信されたXMLデータ13c、およびこのXMLデータ13cの文書型を定義したDTDファイル13d等が格納される。

【0049】グラフィック処理部14には、モニタ14



aが接続されている。このグラフィック処理部14は、CPU11からの命令に従って、モニタ14aの画面上に画像を表示させる。入力I/F15には、キーボード15aやマウス15bが接続されている。この入力I/F15は、キーボード15aやマウス15bからの出力信号を、バス17を介してCPU11に送信する。通信I/F16は、ネットワーク200に接続され、このネットワーク200を介して、商品販売サーバ21および22との間でデータの送受信を行う。

【0050】以上のようなハードウェア構成によって、本実施の形態の処理機能を実現することができる。具体的には、受注管理プログラム13aおよびメモリ展開プログラム13bが、RAM12上にロードされ、CPU11によって実行されることにより実現される。

【0051】ここで、受注管理プログラム13aは、上述したように商品を受注してその受注状況を管理する処理を行うプログラムである。この受注管理プログラム13aが実行されると、商品販売サーバ21または22からネットワーク200を通じて送信された、商品を受注するためのXMLデータ13cが、通信I/F16により受信され、このXMLデータ13cをRAM12上に展開するように、メモリ展開プログラム13bに対して展開要求を行う。このXMLデータ13cのRAM12上での展開により、受注管理プログラム13aは受信したXMLデータ13cの要素データにアクセスし、これを利用して商品名や購入者情報等の発注内容を認識することが可能となる。

【0052】また、メモリ展開プログラム13bは、例えば受注管理プログラム13aの起動と同時に起動される。メモリ展開プログラム13bは、XMLパーサと呼ばれるプログラムの一部として構成され、通信I/F16を介して受信されたXMLデータ13cをRAM12上に展開することにより、メモリ展開プログラム13b側がXMLデータ13cの内容を読み取ることを可能にする。

【0053】ここで、従来のXMLデータ13cのRAM12への展開処理では、受注管理プログラム13aからの展開要求を受信するたびに、RAM12上に要素データを格納するためのメモリ領域を確保して要素データを格納し、XMLデータ13cの有するツリー構造に従って、各メモリ領域を関連付ける処理を行っていた。

【0054】しかし、受信されるXMLデータ13cのツリー構造は、通常、DTDファイル13dによってそのほぼ全容があらかじめ定義されているため、展開要求が発せられた時点では確保すべきメモリ構造がほとんど判明している。本発明ではこの点に着目して、DTDファイル13dによる定義に基づいて、要素データを格納するメモリ領域の確保とその関連付けの処理をあらかじめ行い、展開要求の受信後には確保したメモリ領域に要素データを格納する処理のみを行うようにすることによ

り、受注管理プログラム13aからの展開要求後の処理を高速度化して、そのレスポンスを高める。

【0055】次に、受注管理プログラム13aによって処理される受注対象の商品として本を例に挙げ、商品販売サーバ21および22から商品受注サーバ10に対して送信されるデータの具体例について説明する。商品販売サーバ21および22からは、端末装置31、32および33等のユーザからの注文を受けた本の注文伝票を、XMLデータ13cとして商品受注サーバ10に対して送信する。このXMLデータ13cの文書構造はDTDファイル13dにより定義される。

【0056】ここで、図4は、本の注文伝票の文書構造を定義するDTDファイル13dのプログラムリスト例を示す図である。図4に示すプログラムリストでは、第401行目において、ルート要素として本の注文伝票であることを示す「本注文」が定義され、その子要素として「発注元」「タイトル」の2つが定義されている。ここで、「発注元」は、このDTDファイル13dに基づくXMLデータ13cの送信元である商品販売サーバ21または22において、本の注文を受けた相手、すなわち端末装置31、32および33のいずれかのユーザについての情報を示し、「タイトル」はそのユーザにより注文された本のタイトルを示している。

【0057】次に、第402行目において、「発注元」の子要素として、発注したユーザについての「氏名」「住所」という2つの要素が定義されている。そして、第403行目および第404行目において、末端の要素である「氏名」「住所」に対してそれぞれ文字データを格納することが定義されている。また、同様に第405行目でも、「タイトル」の要素に対して文字データを格納することが定義されている。

【0058】このように、DTDファイル13dは、XMLデータ13cの具備するツリー構造を定義する。DTDファイル13dは、商品受注サーバ10とともに取引相手である商品販売サーバ21および22にも同一のものが保持される。商品販売サーバ21および22は、端末装置31、32および33のユーザからの注文を受けると、DTDファイル13dを参照し、各装置から受信したデータを基に、「氏名」「住所」「タイトル」の各要素について文字データを格納したXMLデータを生成して、商品受注サーバ10に対して送信する。

【0059】DTDファイル13dは、例えば取引対象の商品ごとやその種類ごと、また取引形態ごとに、それぞれ異なる文書構造のものが生成されればよい。そして、同一のDTDファイル13dを保持する複数の取引者の間では、統一されたフォーマットのデータを使用した商取引を行うことが可能となる。

【0060】次に、図5は、上記のDTDファイル13dに従って記述された、本の注文伝票のXMLデータ13cのプログラムリスト例を示す図である。図5に示す



プログラムリスト例では、図4に示したDTDファイル13dで定義されたツリー構造に沿って、各要素が記述されている。その中で、第503行目、第504行目、および第506行目において、末端の要素である「氏名」「住所」「タイトル」に対する要素データとしてそれぞれ「name」「address」「title」という文字データが格納されている。

【0061】次に、図6は、メモリ展開プログラム13bの実行によるXMLファイル13cのメモリ展開処理を示すフローチャートである。まず、メモリ展開プログラム13bの起動時、あるいは商品受注サーバ10の起動時、このメモリ展開プログラム13bの商品受注サーバ10へのインストール時等において、初期設定が行われる。ステップS601において、DTDファイル13dを読み込み、受信されるXMLデータ13cに対する文書型の定義情報を取得する。ステップS602において、取得した定義情報に従い、受信するXMLデータ13cを展開するためにRAM12上に確保するメモリ領域の予約数を取得し、これを保持する。

【0062】なお、以上のステップS601およびS602における初期設定が、メモリ展開プログラム13bのインストール時に行われた場合は、この後にメモリ展開プログラム13bが起動される。また、メモリ展開プログラム13bの起動後に、DTDファイル13dが変更されたり、あるいは追加された場合には、新規のDTDファイル13dについて、上記のステップS601およびS602における処理をその都度行う。この場合の処理は、後述するステップS604～S608までの処理が行われていない状態、すなわち受注管理プログラム13aからの展開要求の特機時において行われる。

【0063】ステップS603において、ステップS602の処理で保持されたメモリ領域の予約数に基づいて、この後に受信するXMLデータ13cの各要素データを格納するためのメモリ領域をRAM12上に確保し、各メモリ領域をツリー構造となるように関連付ける。これにより、DTDファイル13dによって定義された文書型のツリー構造に対応するようにメモリ領域が関連付けられた構造化メモリが、RAM12上に確保される。

【0064】なお、このとき複数の構造化メモリをRAM12上に確保してもよい。例えば、異なる文書型が定義された複数のDTDファイル13dが存在する場合は、各DTDファイル13dの文書型に対応する構造を有する構造化メモリを確保しておく。また、同一の構造を有する構造化メモリを複数確保してもよい。

【0065】ステップS603の後、商品管理サーバ21および22から商品発注のためのXMLデータ13cが送信され、商品受注サーバ10では、通信I/F16においてこのXMLデータ13cが順次受信され、HD13上に記憶される。ステップS604において、記

憶したXMLデータ13cのそれぞれについて、受注管理プログラム13aによってRAM12上への展開要求が順次行われ、メモリ展開プログラム13bはこの展開要求を受信する。

【0066】ステップS605において、展開対象となるXMLデータ13cを解析して、データが格納された要素を1つ取り出す。ステップS606において、確保された構造化メモリ中の、XMLデータ13cにおけるツリー構造に対応するメモリ領域に、取り出した要素の要素データを設定する。このとき、ステップS603の処理で確保された構造化メモリのうち、データが格納されていない、または格納されたデータを保持する必要がなくなった空きメモリを選択して、この中のメモリ領域に要素データを設定していく。

【0067】ステップS607において、展開対象のXMLデータ13cに含まれるすべての要素データに対する展開が終了したか否かを判断し、終了した場合はステップS608に進む。また、終了していない場合はステップS605に戻り、次の要素を取り出して要素データのメモリ領域への格納を繰り返す。

【0068】ステップS608において、すべての要素データに対する展開が終了すると、展開済みのメモリ領域についての情報を受注管理プログラム13aに対して返却する。これにより、受注管理プログラム13aは、受信したXMLデータ13cに含まれるデータを読み出して、受注された商品や個数、購入者等の情報を認識することが可能となる。

【0069】続くステップS609において、この時点で要素データが格納されている構造化メモリのうち、受注管理プログラム13aによる使用が終了されているものがあるか否かを判断する。このために、例えば、受注管理プログラム13aには、要素データが展開された構造化メモリについて、その要素データを使用中であるか否かを示すフラグデータを出力する機能が備えられており、メモリ展開プログラム13bは、要素データが展開された構造化メモリのそれぞれについて、このフラグデータを監視する。使用が終了した構造化メモリが存在する場合はステップS610に進み、存在しない場合はステップS604に進んで、新たなメモリ展開要求の受信を待機する。

【0070】ステップS610において、受注管理プログラム13aからの使用が終了した構造化メモリについて、そのメモリ領域に格納されている要素データを保持する必要がなくなったことを判断して、そのメモリ領域をリフレッシュし、新たなXMLデータ13cの展開が可能な空きメモリとして設定する。このとき、メモリ領域のリフレッシュを行わずに、該当する構造化メモリに対して新たなXMLデータ13cの展開が可能であることを認識しておくだけでもよい。このように、受注管理プログラム13aからの使用が終了して再び空きメモリ

として設定された構造化メモリは、この後ステップS604に戻って再びメモリ展開要求が受信された際に、新たなXMLデータ13cを受け入れることが可能となる。

【0071】なお、構造化メモリが複数確保されている場合は、上記のステップS608の後から、ステップS604に戻って再びメモリ展開要求を受けるまでの待機時間中に、構造化メモリの格納データの使用が終了されたか否かを監視するステップS609での処理を、例えば定期的に行って、使用終了が検知された場合はその都度、該当する構造化メモリを空きメモリに設定するステップS610での処理を行う。

【0072】また、例えば、ステップS604において新たなメモリ展開要求を受信した際に、空きメモリに設定された構造化メモリが存在しない場合は、メモリ展開要求に応じて新たな構造化メモリの確保を行ってから要素データの展開を行うか、あるいは、データが展開された構造化メモリが開放されるまで、新たなXMLデータ13cの展開処理の実行待機状態とする必要がある。例えば、ステップS607で全データのメモリ展開が終了した後に、空きメモリの数を検知し、1つも残っていない場合はその時点で新たな構造化メモリを確保するようにしてもよい。

【0073】次に、図7は、図6のメモリ展開処理に伴って確保される構造化メモリの状態遷移例を模式的に示す図である。図7では、図4で示したDTDファイル13dが参照されて、受信したXMLファイル13cがRAM12上に展開されたときに確保される構造化メモリの状態が示されている。

【0074】図7(A)は、メモリ展開プログラム13bの起動時、あるいは商品受注サーバ10の起動時、このメモリ展開プログラム13bの商品受注サーバ10へのインストール時等において、メモリ展開プログラム13bの初期設定に基づいてあらかじめ確保される構造化メモリの状態を示している。この状態は、図6のステップS602での処理後の状態に相当する。ここでは、参照したDTDファイル13dに従って、メモリ領域が同一のツリー構造となるように対応付けられる。このツリー構造において、受信するXMLデータ13cから要素データが格納される「氏名」「住所」「タイトル」の各要素については、メモリ領域が確保されるだけで、データは空の状態となっている。

【0075】次に、(B)は、受注管理プログラム13aからのメモリ展開要求を受信して、受信されたXMLデータ13cを解析してすべての要素データを順次取り出し、それぞれを対応するメモリ領域に格納したときの状態を示している。この状態は、図6のステップS607で全データのメモリ展開が終了したときの状態に相当する。ここでは、あらかじめ確保されていた「氏名」「住所」「タイトル」の各要素に対するメモリ領域にそ

れぞれ文字データが格納されている。この後、受注管理プログラム13aはこの構造化メモリにアクセスし、格納された文字データを使用して受注処理を行う。

【0076】次に、(C)は、受注管理プログラム13aによる要素データの使用が終了し、メモリ領域に格納されたデータを保持する必要がなくなったときの状態を示している。この状態は、図6のステップS610の処理後の状態に相当する。ここでは、確保されたツリー構造はそのまま、文字データが格納されていたメモリ領域がリフレッシュされている。これにより、この後に再び受注管理プログラム13aからのメモリ展開要求を受信したときに、取り出した文字データをリフレッシュされたメモリ領域に再び格納することで、新たな構造化メモリを確保せずにメモリ展開処理を行うことが可能となる。

【0077】以上のメモリ展開処理では、受注管理プログラム13aによるメモリ展開要求に先立って、初期設定の段階でDTDファイル13dに基づくツリー構造にメモリ領域を関連付けた構造化メモリをあらかじめ確保している。すなわち、メモリ領域の確保とそれをツリー構造とするための関連付けをあらかじめ済ませている。これにより、メモリ展開要求を受信したときには、XMLデータ13cから取り出した要素データを、確保しているメモリ領域に設定する処理のみ行う。DTDファイル13dを参照することにより、構造化メモリでは受信したXMLデータ13cのデータ構造にマッチしたメモリ構造が確保され、要素データの設定が効率よく行われる。従って、メモリ展開要求後の処理を高速化することができ、展開性能が向上する。

【0078】また、例えば、受注管理プログラム13aがマルチスレッドで動作する場合は、各スレッドからのメモリ展開要求を受信するたびに、メモリ領域の確保および関連付けを行う必要がなくなるので、処理速度が高められる。

【0079】さらに、一度データが展開された構造化メモリは、データの使用が終了すると空きメモリに設定することにより、確保したメモリ領域を残したまま再使用することができるので、その後にメモリ展開要求を受信した場合の処理速度を低下させることがない。また、空きメモリの設定処理は、必ずメモリ展開要求に対する待機状態において行うことで、CPUリソースを使用するタイミングが分散されて処理効率が高まり、メモリ展開時の速度を低下させずに済む。

【0080】また、万が一空きメモリが1つも存在しなくなった場合には、メモリ展開要求の待機状態において、空きメモリの数を監視し、足りない場合にはこの時点で新たな構造化メモリを確保しておくことで、メモリ展開要求後にあらためて構造化メモリの確保を行う必要がなくなり、処理速度を低下させずに済む。

【0081】ところで、DTDファイル13dによる文

書構造定義では、ある要素について、その子要素の出現回数を特定せずにおくことが可能である。ここで、図8は、このように要素の出現回数が特定されない場合のDTDファイル13dのプログラムリスト例を示す図である。

【0082】図8では、本の注文伝票において、1回の発注で複数の種類の本を注文することが可能な場合のプログラムリスト例を示している。このプログラムリストにおいて、図4に示した例との違いは、「本注文」の要素の子要素を定義している第801行目で、「\*」の記述により「タイトル」の要素をその出現数を定めずに複数記述することが可能であることが示されている。

【0083】次に、図9は、上記の図8のDTDファイル13dに基づいて記述されたXMLデータ13cのプログラムリスト例を示す図である。図9では、1回の発注で「title1」と「title2」の2冊の本が注文された例を示している。このプログラムリストでは、第6行目および第7行目において、注文された本のタイトルを示す文字データとして「title1」と「title2」が記述されている。

【0084】この例のように、DTDファイル13dにおいて出現回数が特定されていない要素が存在する場合は、メモリ展開プログラム13bの起動時にあらかじめ構造化メモリを確保する際、要素データを格納するために確保するメモリ領域の数を確定することができない。そのため、例えば事前に確保しておいたメモリ領域の数より、受信したXMLデータ13cに格納された要素数が多かった場合は、その時点で新たなメモリ領域を確保して関連付けを行わなければならない必要が生じてしまい、処理速度が低下してしまう。

【0085】そこで、実際にメモリ展開処理を行うたびに、不確定な要素についての出現回数を記録して蓄積し、統計的な計算により確保するメモリ領域の数を順次調節する。このような要素の出現回数に対する学習機能を具備することにより、構造化メモリの構造を常に最適化して、メモリ展開要求の受信時点での新たなメモリ領域確保の回数を低減させ、全体として処理速度を高速化することが可能となる。

【0086】ここで、図10は、このように出現回数が不特定な要素が存在する場合のメモリ展開処理を示すフローチャートである。なお、ここでは図6に示したフローチャートに対応する処理については、その説明を一部省略する。

【0087】ステップS1001において、メモリ展開プログラム13cのインストール時または起動時における初期設定を行う。この処理は、図6の処理例のステップS601およびS602に対応し、DTDファイル13dの参照により、出現回数の不確定な要素については、メモリ領域の予約数として所定の初期値を設定する。また、その予約数を実際の要素の出現回数で最適化

するための計算式（ステップS1010で使用）も設定される。ステップS1002において、上記の初期設定に基づいて、RAM12上にメモリ領域を確保し、ツリー構造となるように関連付けを行う。

【0088】ステップS1003において、受注管理プログラム13aからのメモリ展開要求を受信する。ステップS1004は、図6のステップS605～S608の処理に対応し、受信されたXMLデータ13cから取り出した要素データが、確保されていたメモリ領域に順次格納する処理が行われる。ここで、取り出された要素データの数が確保したメモリ領域の数より多い場合は、あらためてメモリ領域を追加確保し、親要素との関連付けを行う処理が行われる。また、逆に要素データの数の方が少ない場合は、空きのままのメモリ領域が存在することになる。

【0089】ステップS1005において、ステップS1004の処理で、出現回数が不確定な要素の展開が行われたか否かを判断し、展開されていた場合はステップS1006に進み、展開されていなかった場合はステップS1007に進む。ステップS1006において、出現回数が不確定な要素が、ステップS1004での展開処理で実際に出現した回数を記憶する。

【0090】ステップS1007およびS1008は、図6のステップS609およびS610の処理に対応し、受注管理プログラム13aによる使用が終了し、格納された要素データを保持する必要がなくなった構造化メモリがある場合には、この構造化メモリを空きメモリとして設定する。

【0091】ステップS1009において、上記のステップS1004の処理で、出現回数が不確定な要素の展開が行われたか否かを再び判断し、展開されていた場合はステップS1010に進み、展開されていなかった場合はステップS1003に進んで、再びメモリ展開要求を受信する。

【0092】ステップS1010において、出現回数が不確定な要素について、上記のステップS1006で記憶されていた実際の出現回数の蓄積数から、この要素についてのメモリ領域の最適な予約数を統計的に算出する。このときの計算式としては、例えば、蓄積された過去の要素の出現回数の合計値を、過去のメモリ展開回数（ステップS1004の実行回数）で除算して、出現回数の平均値を算出する等が適用可能である。ステップS1011において、算出された統計値に基づいて、構造化メモリの有するメモリ領域の確保数を変更し、ツリー構造を最適化する。

【0093】以上の処理では、不確定な要素の出現回数に対する学習機能を具備することにより、メモリ展開要求の受信時点での新たなメモリ領域確保の回数が低減し、処理速度を総じて高速化することが可能となる。また、この学習機能による構造化メモリの構造変更処理

は、メモリ展開要求に対する待機状態で行われるので、要求後のメモリ展開処理に影響を与えず、CPUリソースを有効に利用することができる。

【0094】なお、以上のような処理内容を記述したメモリ展開プログラム13cは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリなどがある。磁気記録装置には、HDD、FD（フレキシブルディスク）、磁気テープなどがある。光ディスクには、DVD(Digital Versatile Disk)、DVD-RAM、CD-ROM(Compact Disk-Read Only Memory)、CD-R(Recordable)/RW(Rewritable)などがある。光磁気記録媒体には、MO(Magneto-Optical disk)などがある。

【0095】プログラムを流通させる場合には、たとえば、そのプログラムが記録されたDVD、CD-ROMなどの可搬型記録媒体が販売される。また、プログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することもできる。

【0096】プログラムを実行するコンピュータは、たとえば、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、自己の記憶装置に格納する。そして、コンピュータは、自己の記憶装置からプログラムを読み取り、プログラムに従った処理を実行する。なお、コンピュータは、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。また、コンピュータは、サーバコンピュータからプログラムが転送される毎に、逐次、受け取ったプログラムに従った処理を実行することもできる。

【0097】(付記1) あらかじめ文書構造定義が行われたツリー構造データをメモリ上に展開する処理をコンピュータに実行させるメモリ展開プログラムにおいて、前記文書構造定義に基づいてメモリ領域をツリー構造に関連付けた構造化メモリをあらかじめ確保し、前記ツリー構造データについての展開要求を受信すると、読み込んだ前記ツリー構造データに含まれる要素データを前記構造化メモリ中の対応する前記メモリ領域に格納する、処理を前記コンピュータに実行させることを特徴とするメモリ展開プログラム。

【0098】(付記2) 前記構造化メモリをあらかじめ複数確保しておき、前記展開要求を行うスレッドが複数存在する場合に、確保されたうちの前記要素データの格納が行われていないいずれかの前記構造化メモリを前記各スレッドが任意に使用可能とされることを特徴する付記1記載のメモリ展開プログラム。

【0099】(付記3) 前記構造化メモリに格納された前記要素データを保持しておく必要がなくなった場合

は、前記展開要求に対する待機状態において、該当する前記構造化メモリを新たな前記ツリー構造データの格納処理が可能な空きメモリとして設定することを特徴とする付記1記載のメモリ展開プログラム。

【0100】(付記4) 前記展開要求に対する待機状態において、必要に応じて前記構造化メモリを追加確保することを特徴とする付記1記載のメモリ展開プログラム。

(付記5) 前記文書構造定義において出現回数が不特定の要素が設定されている場合に、前記ツリー構造データの前記構造化メモリへの格納処理を行うたびに前記要素の実際の出現回数を記憶して順次蓄積し、蓄積した値に基づいて必要な前記メモリ領域の数を統計的に算出し、前記展開要求に対する待機状態において、確保している前記構造化メモリの構造を前記算出値に応じて変更することを特徴とする付記1記載のメモリ展開プログラム。

【0101】(付記6) 前記構造化メモリの確保は、前記メモリ展開プログラムを利用するシステムの起動時、または前記メモリ展開プログラムのインストール時に行われることを特徴とする付記1記載のメモリ展開プログラム。

【0102】(付記7) 前記ツリー構造データは、XML(Extensible Markup Language)に従って記述されることを特徴とする付記1記載のメモリ展開プログラム。

(付記8) あらかじめ文書構造定義が行われたツリー構造データを使用して処理するデータ処理装置において、前記ツリー構造データが展開されるメモリと、前記文書構造定義に基づいてツリー構造に関連付けたメモリ領域を前記メモリ上にあらかじめ確保し、前記ツリー構造データについての展開要求を受信すると、読み込んだ前記ツリー構造データに含まれる要素データを対応する前記メモリ領域に格納するメモリ展開手段と、使用する前記ツリー構造データに対する前記展開要求を前記メモリ展開手段に送信し、前記要素データが格納された前記メモリ領域にアクセスすることにより前記ツリー構造データを読み取って所定の処理を行うツリー構造データ処理手段と、を有することを特徴とするデータ処理装置。

【0103】

【発明の効果】以上説明したように、本発明のメモリ展開プログラムでは、ツリー構造データの要素データを格納するためのメモリ領域を確保する処理やそれらをツリー構造に関連付ける処理が、ツリー構造データのメモリへの展開処理に先立ってあらかじめ行われるので、展開処理時における処理負担が軽減され、処理速度を高速化することが可能となる。

【0104】また、本発明のメモリ展開方法では、ツリー構造データの要素データを格納するためのメモリ領域を確保する処理やそれらをツリー構造に関連付ける処理

が、ツリー構造データのメモリへの展開処理に先立ってあらかじめ行われるので、展開処理時における処理負担が軽減され、処理速度を高速化することが可能となる。

【図面の簡単な説明】

【図1】本発明の原理を説明するための原理図である。

【図2】本発明を適用可能な商品販売システムのシステム構成例を示す図である。

【図3】本発明の実施の形態に用いる商品受注サーバの内部構成例を示すブロック図である。

【図4】本の注文伝票の文書構造を定義するDTDファイルのプログラムリスト例を示す図である。

【図5】図6に示したDTDファイルに従って記述された、本の注文伝票のXMLデータのプログラムリスト例を示す図である。

【図6】メモリ展開プログラムの実行によるメモリ展開処理を示すフローチャートである。

【図7】図6に示したメモリ展開処理に伴って確保される構造化メモリの状態遷移例を模式的に示す図である。

【図8】要素の出現回数が特定されない場合のDTDフ

ァイルのプログラムリスト例を示す図である。

【図9】図8に示したDTDファイルに基づいて記述されたXMLデータのプログラムリスト例を示す図である。

【図10】出現回数が不特定の要素が存在する場合のメモリ展開処理を示すフローチャートである。

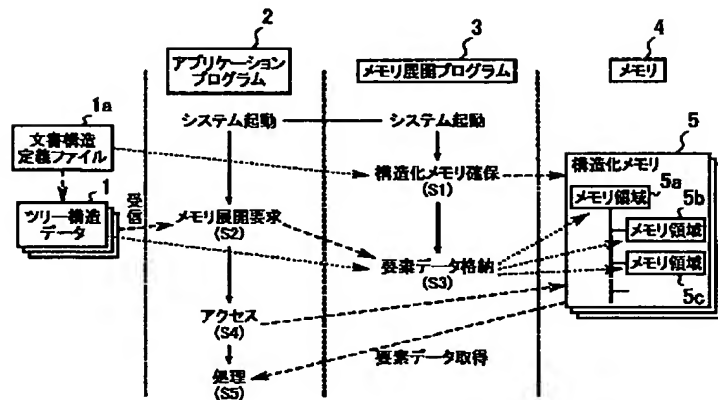
【図11】従来のメモリ展開プログラムによるメモリ展開処理を示すフローチャートである。

【図12】従来のメモリ展開処理により確保されるメモリの状態遷移を模式的に示す図である。

【符号の説明】

- 1 ツリー構造データ
- 1a 文書構造定義ファイル
- 2 アプリケーションプログラム
- 3 メモリ展開プログラム
- 4 メモリ
- 5 構造化メモリ
- 5a、5b、5c メモリ領域

【図1】

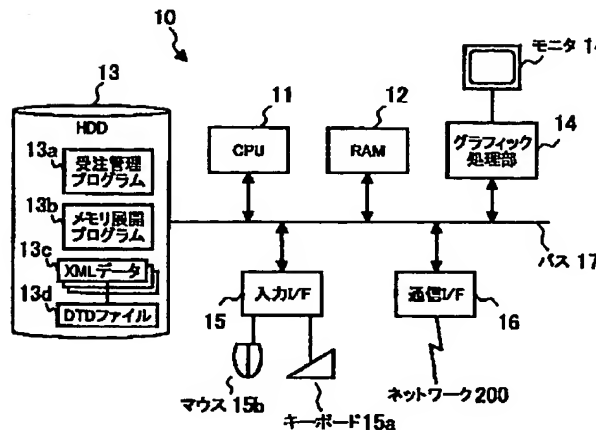


【図4】

- 401 <ELEMENT 本注文(発注元, タイトル)>
- 402 <ELEMENT 発注元(氏名, 住所)>
- 403 <ELEMENT 氏名(#PCDATA)>
- 404 <ELEMENT 住所(#PCDATA)>
- 405 <ELEMENT タイトル(#PCDATA)>

【図9】

【図3】



【図5】

- 501 <本注文>
- 502 <発注元>
- 503 <氏名> name /氏名>
- 504 <住所> address /住所>
- 505 </発注元>
- 506 <タイトル> title /タイトル>
- 507 </本注文>

901 <本注文>

902 <発注元>

903 <氏名> name /氏名>

904 <住所> address /住所>

905 </発注元>

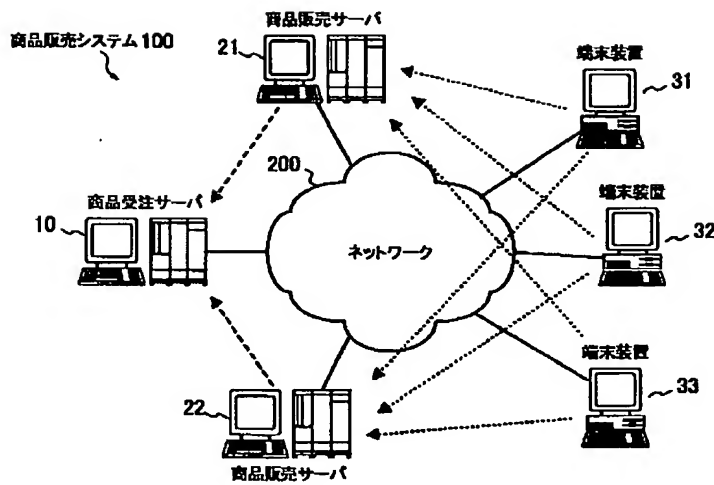
906 <タイトル> title1 /タイトル>

907 <タイトル> title2 /タイトル>

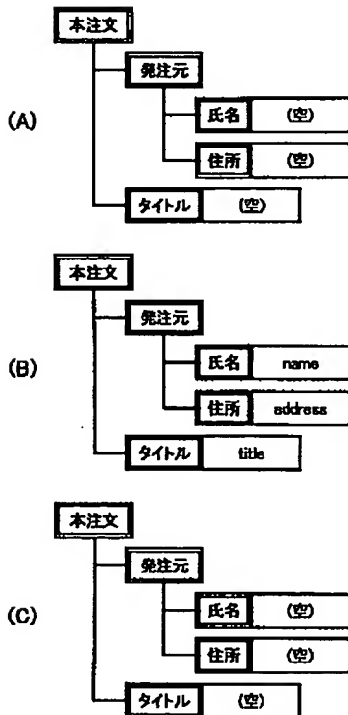
...

xxx </本注文>

【図2】



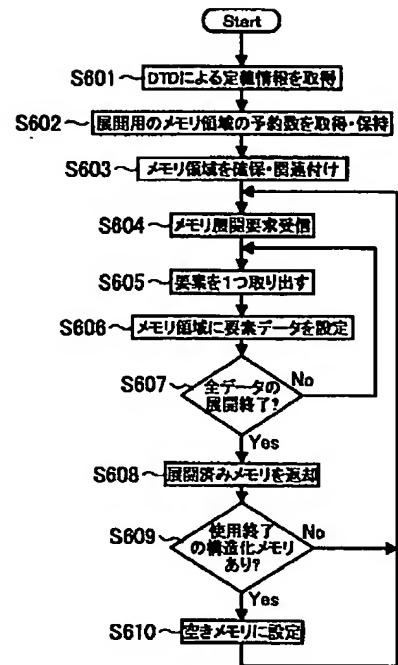
【図7】



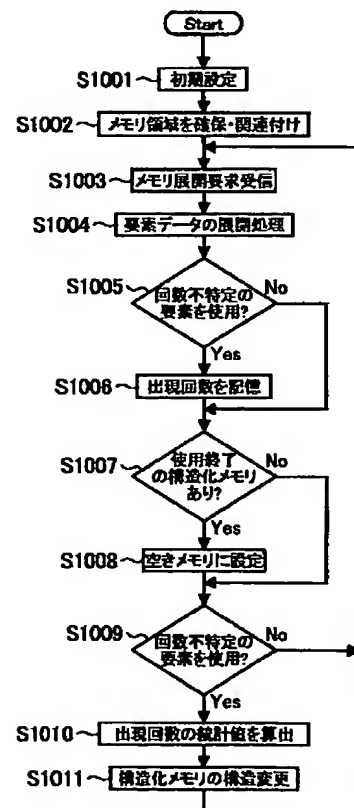
【図8】

- 801 <ELEMENT 本注文 (発注元, タイトル)>  
 802 <ELEMENT 発注元 (氏名, 住所)>  
 803 <ELEMENT 氏名 (#PCDATA)>  
 804 <ELEMENT 住所 (#PCDATA)>  
 805 <ELEMENT タイトル (#PCDATA)>

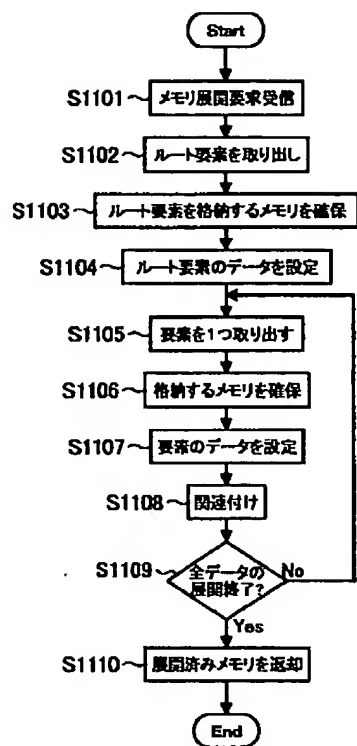
【図6】



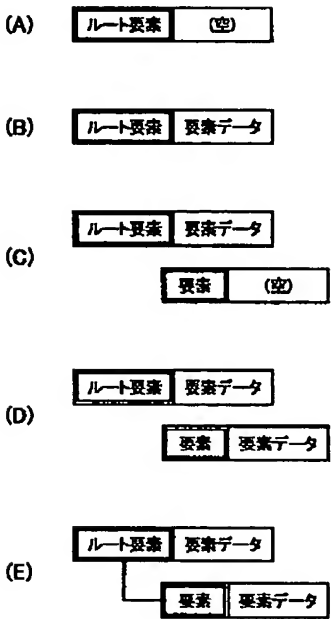
【図10】



【図11】



【図12】



フロントページの続き

(51)Int.Cl. <sup>7</sup>	識別記号	F I	キーワード (参考)
G 0 6 F 17/30	1 6 0	G 0 6 F 17/30	1 6 0 F
	4 1 9		4 1 9 A